Claim Amendments

1. (Original) A computer system, comprising:

a processor equipped to serve as multiple logical processors;

a host chipset connected to the processor;

PCI devices connected to the host chipset, via a PCI bus; and

a main storage connected to the host chipset and arranged to store an operating system (OS) and contain a basic input/output system (system BIOS) configured to execute multiple pre-boot tasks, including memory initialization and PCI bus initialization concurrently, on the processor which serve as multiple logical processors before passing control to the operating system (OS).

2. (Original) The computer system as claimed in claim 1, wherein said main storage comprises:

a main memory arranged to store the operating system (OS) for use by the processor; and

a flash memory arranged to store the system BIOS and other applications that may execute during boot up (start-up) before the operating system (OS) is loaded.

3. (Original) The computer system as claimed in claim 1, wherein said processor equipped to serve as multiple logical processors allows the system BIOS to execute the memory initialization on one of the logical processors while enabling the other logical processor to proceed with PCI bus initialization tasks.

4. (Original) The computer system as claimed in claim 3, wherein said memory initialization is executed by:

detecting a physical memory array plugged into the computer system;

programming the host chipset with the physical memory specifics, and initializing the physical memory using the host chipset;

detecting a physical memory ECC (Error Checking and Correction) Capability; and

programming the host chipset with the ECC capability, and writing zeros to the entire physical memory array in order to ensure that memory is usable for the operating system (OS).

5. (Original) The computer system as claimed in claim 1, wherein said processor equipped to serve as multiple logical processors allows the system BIOS to execute the PCI bus initialization on one of the logical processors while enabling the other logical processor to proceed with other normal PCI device initialization tasks.

6. (Original) The computer system as claimed in claim 3, wherein said PCI bus initialization is executed by:

scanning all possible PCI buses connected to the host chipset;

after each PCI bus is scanned, scanning and initializing all possible PCI devices connected thereto on an individual basis;

after each PCI device is scanned, initializing all possible PCI functions assigned thereto on an individual function; and

terminating the PCI bus initialization, when all the PCI buses, all the PCI devices connected to each PCI bus, and all PCI functions assigned to each PCI device are scanned and initialized.

7. (Original) The computer system as claimed in claim 1, wherein said processor equipped to serve as multiple logical processors includes a Boot-Strap Logical Processor (BSLP) assigned to execute a set of pre-boot tasks and one or more

Alternate Logical Processors (ALP) assigned to executed another set of pro-boot tasks concurrently until all assigned pre-boot tasks are completed before passing the control to the operating system (OS).

8. (Original) The computer system as claimed in claim 7, wherein, upon a system reset, said Boot-Strap Logical Processor (BSLP) executes the following pre-boot tasks:

detecting a physical memory, and programming the host chipset;

detecting the presence of all logical processors and sending a SIPI (Startup Inter-Processor Interrupt) to wake up the Alternate Logical Processor (ALP);

initializing all internal hardware of the host chipset, and all storage devices connected to the host chipset; and

waiting for ALP code execution completion to place the Alternate Logical Processor (ALP) in a wait state before passing the control to the operating system (OS).

9. (Original) The computer system as claimed in claim 8, wherein, upon receipt of the SIPI, said Alternate Logical Processor (ALP) executes the memory initialization and the PCI bus initialization respectively, while the Boot-Strap Logical Processor (BSLP) initializes all internal hardware of the host chipset and all storage devices connected to the host chipset, and after the memory initialization and PCI bus initialization are completed, indicates to the Boot-Strap Logical Processor (BSLP) that the memory initialization and PCI bus initialization are completed after all internal hardware of the host chipset and all storage devices connected to the host chipset are initialized by the Boot-Strap Logical Processor (BSLP).

10. (Original) The computer system as claimed in claim 9, wherein said memory initialization is executed by:

detecting a physical memory array plugged into the computer system;

programming the host chipset with the physical memory specifics, and initializing the physical memory using the host chipset;

detecting a physical memory ECC (Error Checking and Correction) Capability; and

programming the host chipset with the ECC capability, and writing zeros to the entire physical memory array in order to ensure that memory is usable for the operating system (OS).

11. (Original) The computer system as claimed in claim 9, wherein said PCI bus initialization is executed by:

scanning all possible PCI buses connected to the host chipset;

after each PCI bus is scanned, scanning and initializing all possible PCI devices connected thereto on an individual basis;

after each PCI device is scanned, initializing all possible PCI functions assigned thereto on an individual function; and

terminating the PCI bus initialization, when all the PCI buses, all the PCI devices connected to each PCI bus, and all PCI functions assigned to each PCI device are scanned and initialized.

12. (Original) A computer system, comprising:

a processor equipped to serve as multiple logical processors;

a host chipset connected to the processor;

PCI devices connected to the host chipset, via a PCI bus;

a main memory arranged to store the operating system (OS) for use by the processor; and

a flash memory arranged to store a basic input/output system (system BIOS) and other applications that may execute during boot up (start-up) before the operating system (OS) is loaded, wherein said system BIOS is configured to execute two different coordinate pre-boot tasks, including memory initialization and PCI bus initialization concurrently, on the processor which serve as multiple logical processors before passing control to the operating system (OS).

13. (Original) The computer system as claimed in claim 12, wherein said processor equipped to serve as multiple logical processors allows the system BIOS to execute the memory initialization on one of the logical processors while enabling the other logical processor to proceed with PCI bus initialization tasks.

14. (Original) The computer system as claimed in claim 13, wherein said memory initialization is executed by:

detecting a physical memory array plugged into the computer system;

programming the host chipset with the physical memory specifics, and initializing the physical memory using the host chipset;

detecting a physical memory ECC (Error Checking and Correction) Capability; and

programming the host chipset with the ECC capability, and writing zeros to the entire physical memory array in order to ensure that memory is usable for the operating system (OS).

15. (Original) The computer system as claimed in claim 12, wherein said processor equipped to serve as multiple logical processors allows the system BIOS to execute the PCI bus initialization on one of the logical processors while enabling the other logical processor to proceed with other normal PCI device initialization tasks.

16. (Original) The computer system as claimed in claim 15, wherein said PCI bus initialization is executed by:

scanning all possible PCI buses connected to the host chipset;

after each PCI bus is scanned, scanning and initializing all possible PCI devices connected thereto on an individual basis;

after each PCI device is scanned, initializing all possible PCI functions assigned thereto on an individual function; and

terminating the PCI bus initialization, when all the PCI buses, all the PCI devices connected to each PCI bus, and all PCI functions assigned to each PCI device are scanned and initialized.

17. (Original) The computer system as claimed in claim 12, wherein said processor equipped to serve as multiple logical processors includes a Boot-Strap Logical Processor (BSLP) assigned to execute a set of pre-boot tasks and one or more Alternate Logical Processors (ALP) assigned to executed another set of pro-boot tasks concurrently until all assigned pre-boot tasks are completed before passing the control to the operating system (OS).

18. (Original) The computer system as claimed in claim 17, wherein, upon a system reset, said Boot-Strap Logical Processor (BSLP) executes the following pre-boot tasks:

detecting a physical memory, and programming the host chipset;

detecting the presence of all logical processors and sending a SIPI (Startup Inter-Processor Interrupt) to wake up the Alternate Logical Processor (ALP);

initializing all internal hardware of the host chipset, and all storage devices connected to the host chipset; and

waiting for ALP code execution completion to place the Alternate Logical Processor (ALP) in a wait state before passing the control to the operating system (OS).

19. (Original) The computer system as claimed in claim 18, wherein, upon receipt of the SIPI, said Alternate Logical Processor (ALP) executes the memory initialization and the PCI bus initialization respectively, while the Boot-Strap Logical Processor (BSLP) initializes all internal hardware of the host chipset and all storage devices connected to the host chipset, and after the memory initialization and PCI bus initialization are completed, indicates to the Boot-Strap Logical Processor (BSLP) that the memory initialization and PCI bus initialization are completed after all internal hardware of the host chipset and all storage devices connected to the host chipset are initialized by the Boot-Strap Logical Processor (BSLP).

20. (Original) The computer system as claimed in claim 19, wherein said memory initialization is executed by:

detecting a physical memory array plugged into the computer system;

programming the host chipset with the physical memory specifics, and initializing the physical memory using the host chipset;

detecting a physical memory ECC (Error Checking and Correction) Capability; and

programming the host chipset with the ECC capability, and writing zeros to the entire physical memory array in order to ensure that memory is usable for the operating system (OS).

21. (Original) The computer system as claimed in claim 19, wherein said PCI bus initialization is executed by:

scanning all possible PCI buses connected to the host chipset;

after each PCI bus is scanned, scanning and initializing all possible PCI devices connected thereto on an individual basis;

after each PCI device is scanned, initializing all possible PCI functions assigned thereto on an individual function; and

terminating the PCI bus initialization, when all the PCI buses, all the PCI devices connected to each PCI bus, and all PCI functions assigned to each PCI device are scanned and initialized.

22. (Original) A computer readable medium having stored thereon a set of system basic input/output start-up "system BIOS" instructions configured a single physical processor equipped to serve as a Boot-Strap Logical Processor (BSLP) and an Alternate Logical Processor (ALP) which, when executed by a processor during start-up, cause the Boot-Strap Logical Processor (BSLP) and the Alternate Logical Processor (ALP) to perform:

detecting, at the Boot-Strap Logical Processor (BSLP), detecting a physical memory, programming the host chipset, detecting the presence of all logical processors and sending a SIPI (Startup Inter-Processor Interrupt) to wake up the Alternate Logical Processor (ALP);

initializing, at the Boot-Strap Logical Processor (BSLP), all internal hardware of the host chipset, all storage devices connected to the host chipset, and waiting for ALP code execution completion to place the Alternate Logical Processor (ALP) in a wait state before passing the control to the operating system (OS); and

executing, at the Alternate Logical Processor (ALP), memory initialization and PCI bus initialization respectively, while the Boot-Strap Logical Processor (BSLP) initializes all internal hardware of the host chipset and all storage devices connected to

the host chipset, and after the memory initialization and PCI bus initialization are completed, indicates to the Boot-Strap Logical Processor (BSLP) that the memory initialization and PCI bus initialization are completed after all internal hardware of the host chipset and all storage devices connected to the host chipset are initialized by the Boot-Strap Logical Processor (BSLP).

23. (Original) The computer readable medium as claimed in claim 22, wherein said memory initialization is executed by the Alternate Logical Processor (ALP) by:

detecting a physical memory array plugged into the computer system;

programming the host chipset with the physical memory specifics, and initializing the physical memory using the host chipset;

detecting a physical memory ECC (Error Checking and Correction) Capability; and

programming the host chipset with the ECC capability, and writing zeros to the entire physical memory array in order to ensure that memory is usable for the operating system (OS).

24. (Original) The computer readable medium as claimed in claim 22, wherein said PCI bus initialization is executed by the Alternate Logical Processor (ALP) by:

scanning all possible PCI buses connected to the host chipset;

after each PCI bus is scanned, scanning and initializing all possible PCI devices connected thereto on an individual basis;

after each PCI device is scanned, initializing all possible PCI functions assigned thereto on an individual function; and

terminating the PCI bus initialization, when all the PCI buses, all the PCI devices connected to each PCI bus, and all PCI functions assigned to each PCI device are scanned and initialized.

25. (Currently Amended) A system, comprising:

a memory comprising first instructions associated with a first startup initialization task and second instructions associated with a second startup initialization task; and

a processor comprising a first logical processor to execute the first instructions associated with the first startup initialization task and a second logical processor to initiate execution of the second instructions associated with the second startup initialization task prior to completing execution of the first instructions.

26. (Canceled).

27. (Original) The system as claimed in claim 25, further comprising a second memory, wherein the first instructions in response to being executed result in the processor initializing the second memory.

28. (Currently Amended) The system as claimed in claim 25, further comprising an error checking and correction ~~code~~ (ECC) memory, wherein the second instructions in response to being executed result in the processor initializing the ~~error-correction code~~ ECC memory.

29. (Original) The system as claimed in claim 25, further comprising a peripheral bus and associated devices, wherein the first instructions in response to being executed result in the processor initializing the peripheral bus and associated devices.

30. (Original) The system as claimed in claim 25, further comprising a peripheral component interconnect bus and associated devices, wherein the first instructions in

response to being executed result in the processor initializing the peripheral component interconnect bus and associated devices.

31. (Currently Amended) A method, comprising:

executing with a first logical processor of a processor a first startup initialization task; and

executing with a second logical processor of the processor at least a portion of a second startup initialization task concurrently with execution of the first startup initialization task.

32. (Canceled).

33. (Original) The method as claimed in claim 31, wherein executing the second startup initialization task comprises initializing a memory.

34. (Currently Amended) The method as claimed in claim 31, wherein executing the first startup initialization task comprises initializing an error checking and correction (ECC) ~~code~~ memory.

35. (Original) The method as claimed in claim 31, wherein executing the second startup initialization task comprises initializing a peripheral bus and associated devices.

36. (Original) The system as claimed in claim 25, wherein executing the first startup initialization task comprises initializing a peripheral component interconnect bus and associated devices.

37. (Currently Amended) A computer readable medium comprising first instructions associated with a first hardware initialization task and second instructions associated with a second hardware initialization task which, in response to being executed by a processor, result in the processor

executing a first hardware initialization task with a first logical processor of the processor; and

executing at least a portion of the second hardware initialization during the first hardware initialization task with a second logical processor of the processor.

38. (Canceled).

39. (Currently Amended) The computer readable medium as claimed in claim 38, wherein the first instructions, in response to being executed, further result in the first logical processor initializing an error checking and correction (ECC) ~~code~~ memory.

40. (Original) The computer readable medium as claimed in claim 39, wherein the second instructions, in response to being executed, further result in the second logical processor initializing a peripheral component interconnect bus and associated devices.